**УДК 681.3.06**

# ЭКСПЛИКАТИВНЫЙ БАЗИС ИНТЕГРАЦИОННЫХ СРЕД

*Редько И.В.*

Национальный техничкский университет Украины "КПИ", факультет электроники
01037, Киев, пр. Победы, 37, тел. 441-10-60
sms@ipnet.kiev.ua

Рассматривается понятие дескриптивной среды как системы взаимодействия трех сред – дескриптологической, предметной и интеграционной. Дается экспликация понятий интеграционной среды. На этой основе строится полная система соответствующих дескриптологических операций. Рассматриваются репрезентативные примеры. Библиогр.: 13 назв.

The notion of descriptive environment as a system of interaction of descriptological, integration and data environments is considered. The notion of integration environment is explained. On this basis the complete set of suitable descripological operations are constructed. Then the representative examples of utilization of them are considered in this environment. Refs.: 13 titles.

## Введение

Интенсивное развитие информационных технологий сместило акценты с решения отдельных задач на создание концептуально-единых сред, поддерживающих их интеграцию в рамках комплексной проблематики. Все это выдвинуло на передний план не столько сущности (объекты, субъекты, явления и т.д.), участвующие в решении локальных задач, сколько процессы, в которых эти сущности участвуют. Поэтому процессональная парадигма стала доминирующей в современных средах программирования и решения задач. Однако, невзирая на несомненные преимущества процессонального подхода по сравнению с функциональным, были обнаружены и принципиальные недостатки первого. Излишняя общность его вошла в противоречие с недостаточной содержательностью, проявившейся в системах реального решения задач. Это привело к необходимости обогащения процессональной среды. В связи с этим в ней была индивидуализирована (выделена) *дескриптивная среда* (ДС), поддерживающая процессы дескриптирования (лат. descriptio – описание) сущностей, в том числе сущностей типа самих процессов. Настоящая работа посвящена интеграционным составляющим этой среды. Все используемые и не определяемые в работе понятия и результаты понимаются в смысле [1].

## Общий взгляд на интеграционную среду

В первом приближении интеграционные среды (ИС) представляют собой интерфейсные средства между составляющими ДС – дескриптологической и предметными средами. Роли их хотя и принципиально разные, однако подчинены единой цели – эволюционному созданию и развитию самой ДС. Эволюционность здесь понимается в смысле необходимости эффективного решения проблемы сохранения инвестиций как результатов развития традиционных платформ, при решении основной задачи. Кратко охарактеризуем как упомянутые среды, так и их роли.

Развитие любой среды, в частности и ДС, обязательно предполагает как создание соответствующей понятийной структуры, так и единой логики ее дальнейшего использования. Причем для обеспечения концептуального единства такой среды, ее понятийная структура должна быть в свою очередь производной единственного основополагающего понятия. Так, например, создание и эффективное использование как теоретико-множественной платформы, так и развитой на ее базе среды обеспечено их концептуальным единством. Последнее базируется как на единой категории – понятии множества, так и на соответствующей ему логике. Ведь именно наличие этих двух составляющих позволило естественным образом как вовлечь на концептуально-единой основе в рассмотрения результаты математики, тем самым, разрешив проблему сохранения инвестиций, так и обеспечить дальнейшее их качественное развитие на основе единой теоретико-множественной логики. Вопросы интенсионального развития традиционных платформ, в частности теоретико-множественной платформы и создания на этой основе упомянутой дескриптологической среды обстоятельно изложены в работах ([1-3] и библиографии к ним). Поэтому здесь ограничимся лишь беглым изложением принципиальных моментов. Что касается понятийной структуры *дескриптологической среды* (ДЛС), то она наряду с такими важнейшими понятиями как акция, композиция, действие, собрание, полиада и некоторыми другими, включает в себя также и прагматико-мотивированные связями между ними. Понятийным ядром этой среды является понятие акции, естественным образом интегрирующие в себе статику традиционных собраний, в частности множеств и динамику сущностей типа действий. Логика же среды индуцирована упомянутыми прагматико-мотивированными связями между основными понятиями и практически является их адекватным уточнением.

Вскрытие логической составляющей ДС это важнейший, однако, не единственный шаг в ее экспликации. Не меньшую значимость здесь представляют *предметные среды* (ПС). Ведь именно возможность вовлечения в рассмотрения предметных результатов на основе единой логики и их дальнейшее развитие является основным смыслом построения самой ДС. При этом ПС могут быть в свою очередь сколь угодно сложно устроены. Поэтому обобщающее рассмотрение их с точки зрения открыто-замкнутых систем является на сегодняшний день преждевременным. Разумный путь здесь – основываясь на анализе развитых и прошедших проверку временем ПС по возможности вскрывать их денотативную логику, базирующуюся на достаточно развитой коннотативной логике. В этом смысле ПС представляет собой систему взаимодополняющих друг друга составляющих – *предметную платформу* (ПП), основу которой составляет коннотативная логика, и открытую систему средств денотативной логики. Таким образом, последняя развивает ПП до ПС через вскрытие общезначимых правил использования коннотативных средств. Что касается конкретных предметных сред, то остановимся на репрезентативной среди них – среде программирования [3] и тесно связанными с ней средами именных, $n$-арных (кортежных), абстрактных функций и функций как правил, в частности функций переменных. Здесь получены зримые результаты в области вскрытия коннотативной логики программирования как процесса построения программ. Вскрытие же соответствующей денотативной логики будет здесь в основном сведено к формальной экспликации различных механизмов циклирований, в частности известного содержательного понятия инварианта цикла, в виде общего понятия редукции и развитие на этой основе метода редукций и направления экспликативного моделирования в целом ([4-6] и библиографии к ним). При этом отметим еще раз, что аппарат редукций, хотя и является весьма мощным, все же не образует прагматически полную систему. В этом смысле базирующаяся на нем *интеграционно-предметная среда* (ИПС) является открытой для пополнения ее новыми видами редукций. Такое положение есть следствием того, что основным требованием, предъявляемым к редукционным средствам, является транслируемость редукционных спецификаций в коннотативные (композиционные) спецификации. Это существенно ограничивает выразительные возможности первых. В виду этого первые, как и последние, вообще говоря, не поддерживают средств квантификации. Уже это, в соответствии с принципом достаточных оснований Лейбница ([7], стр. 212, 234, 418, 564), делает принципиально невозможным ограничиться в рассмотрениях денотативной логики какой-либо замкнутой системой редукционных средств. Проблема же транслируемости обогащенных средствами квантификации редукционных спецификаций в общем случае алгоритмически неразрешима. Разумный путь здесь – во-первых, ограничиваться поиском хотя и частных, но прагматически важных случаев таких обогащенных спецификаций, для которых проблема транслируемости разрешима. И, во-вторых, продолжать поиск подходящих типов редукций для важных с прагматической точки зрения случаев коннотативных спецификаций. В этих направлениях пока сделаны только первые шаги.

И, наконец, последний компонент дескриптивной среды – ИС. Основополагающая роль ИС состоит в обеспечении интерфейса между предметными средами, в частности традиционной теоретико-множественной средой, и интенсиональным развитием последних – дескриптологической средой. Это существенно отличает дескриптивную среду от традиционных сред. Ведь в них специальная интеграционная среда чаще всего явно не выделяется. И хотя обеспечение интерфейса необходимо и здесь, специфика задач, на которые ориентированы такие среды, не требуют обеспечения автоматического их (интерфейсных задач) решения. Последнее здесь осуществляется «в розницу», человеком. Дескриптивная же среда, как уже отмечалось, ориентирована в первую очередь на задачи, связанные с программированием и в целом с информатикой. Автоматичность их решений является одним из основных требований, предъявляемых к ним. Отсюда необходимость явного выделения и адекватного уточнения в дескриптивной среде упомянутой интеграционной среды как средства обеспечения преемственности традиционных результатов в интенсиональных рассмотрениях. Основу ее составляют операции введения и исключения абстракции, представляющие собой преобразования относительно конкретных типов сущностей в более абстрактные и наоборот, а также их прагматико-мотивированные пошаговые применения.

## Интеграционный базис дескриптивной среды

Как уже отмечалось, развитие интеграционной среды связано с определением интерфейсных механизмов между традиционными предметными и нетрадиционной дескриптологической средами. Фундаментальным нетрадиционным понятием дескриптологической среды есть понятие акции [1, 3, 4]. Поэтому средства интеграционной среды должны предоставлять механизмы переходов как от понятий традиционных сред к понятиям акции и ассоциированным с ним понятиям, так и наоборот. Для большинства традиционных понятий, таких как множество, упорядоченная пара, именное множество, кортеж и др., связь их с понятиями акциональной среды – собранием, действием, полиадой и т.п. достаточно очевидна. Поэтому здесь ограничимся рассмотрением интеграционных механизмов на репрезентативном примере реализации интерфейса между фундаментальными понятиями: традиционным – функцией и нетрадиционным – акцией. Непосредственная связь между ними весьма неочевидна. Ведь, как следует из определения понятия акции, оно существенно богаче понятия функции. Кроме того, на сегодня в традициях в самостоятельное рассмотрение выделено достаточно много различных видов единого родового понятия функции. Такие рассмотрения взаимодополняют друг друга, делая более содержательным общее понятие функции. Поэтому естественным представляется, наряду с эксплицированием связи между общим понятием функции и понятием акции, пошаговое построение серии дополнительных интерфейсов между традиционными видами функций и соответствующими им универсумами акций. Ряд этих интерфейсов и соответствующих им понятий, на наш взгляд, целесообразно построить так: функция $\leftrightarrow$кортежная ( $n$ -арная) функция $\leftrightarrow$ именная функция $\leftrightarrow$ акция. Первый шаг в этом направлении – целенаправленное обогащение универсума акций $A$ путем индивидуализации в нем специальных универсумов акций – интенсиональных прообразов упомянутых традиционных понятий. Для этого напомним несколько необходимых определений.

Отталкиваясь от понятия действия (см. [1, 3, 4] и библиографии к ним), сущности, абстрагированные (отвлеченные) от действий будем называть *статическими сущностями*. Тогда акции, предпосылки и последствия терминальных действий которых суть статические сущности договоримся называть *акциями функционального* (*статического*) *типа*. Универсум таких акций обозначим через $F$ . Если все действия акции статического типа терминальные, то акцию будем называть *тотальной*, иначе – *частичной акцией*. Очевидно, что такое обогащение универсума $A$ путем введения понятий акции функционального типа, а также его видов – тотальной и частичной акций обеспечивает адекватное интенсиональное уточнение понятий функции, тотальной функции и частичной функции соответственно. При этом заметим, что если случай тотальных функций в определенной степени адекватно отражен в традициях, то случай частичных и тем более общий случай функций в принципе не могут быть адекватно представлены там ввиду их существенно динамической природы. В этом смысле наше толкование этих понятий адекватно обогащает традиционные представления о них. В качестве интерфейсных механизмов здесь достаточно определить операции введения и исключения абстракции между типами абстракций функции и акции статического типа (статической акции) $Ab$ и $Con$ . Введение абстракции $Ab$ является монадной акцией, которая сопоставляет любой функции $f: S \to Q$ соответствующую ей статическую акцию действий сопоставления $Ab(f)$ . Предпосылками таких действий выступают статические сущности $s \in S$ , а последствиями – статические сущности вида $f(s) \in Q$ . Исключение абстракции $Con$ представляет собой монадную акцию, сопоставляющую любой акции статического типа $\alpha$ функцию $Con(\alpha)$ как закон сопоставления любому элементу $p$ из собрания (множества) всех предпосылок действий из $\alpha$ , обозначаемого нами $\Pr(\alpha)$ , соответствующего ему значения $Ex(\alpha, p)$ из собрания (множества) всех последствий действий из $\alpha$ – $Af(\alpha)$ .

Следующий шаг – обогащение уже универсума тотальных акций $F^t$ . Здесь в виду сказанного, целесообразно выделить в универсуме всех статических сущностей, универсум сущностей типа именных множеств [8-11] и рассмотреть универсум именных акций $F^n$ . Тотальную акцию, предпосылки действий которой суть именные множества договоримся называть *именной акцией*. Так введенное понятие именной акции, очевидно, является адекватной интенсиональной экспликацией известного понятия именной функции. Соответствующие интерфейсные средства экспликацируются здесь уже введенными акциями $Ab$ и $Con$ .

Отталкиваясь от понятия именной функции, легко построить интерфейс их с кортежными функциями как частным случаем первых. Формально он задается монадными акциями $Ab^c$ – введение абстракции кортежной функции и $Con^c$ – исключение абстракции кортежной функции.

Введение абстракции кортежной функции $Ab^c$ это монадная акция, сопоставляющая любой $\{1,2,...,m\}-$ арной функции $f^n$ , $m \in N$ , кортежную функцию $Ab^c(f^n)$ . Последняя строится по $f^n$ путем, во-первых, обобщения любого $\{1,2,...,m\}-$ именного множества $s$ до соответствующего кортежа и во-вторых, сопоставления любому такому кортежу значения $f^n(s)$ .

Исключение абстракции кортежной функции $Con^c$ является монадной акцией, которая сопоставляет кортежной функции $f^c$ именную функцию $Con^c(f^c)$ , строящуюся по $f^c$ через, во-первых, конкретизацию любого кортежа $c$ соответствующим именным множеством со стандартными именами и во-вторых, сопоставление любому такому именному множеству значения $f^c(c)$ .

И наконец, последнее звено нашего ряда – (традиционная) тотальная функция $\leftrightarrow$кортежная функция. По аналогии с предыдущим, под исключением абстракции (традиционной) функции понимаем монадную акцию, сопоставляющую любой тотальной функции $g$ соответствующую унарную функцию $Con^f(g)$ . Введение абстракции $Ab^f$ определяется дуальным образом.

Определением данных групп интерфейсов осуществлена экспликация интеграционной среды в части реализации интерфейса между интенсиональным понятием акции и экстенсиональным понятием функции. Это обеспечивает преемственность всех традиционных теоретико-функциональных результатов в развиваемой нами дескриптивной среде.

## Общий взгляд на интеграционно-предметную среду

ИПС представляет собой денотативную логику ПС. В качестве последней нами выбрана среда программирования как наиболее продвинутая, в смысле вскрытия ее коннотативной (процедурной) логики, репрезентативная среда дескриптивных процессов. Успехи в этом направлении, однако, носят все же достаточно ограниченный характер, так как коннотативная логика непосредственно эксплицирует только средства построения программ, оставляя за рамками рассмотрений процессы их использования. А ведь именно последние составляют суть программирования. В этом смысле процедурная логика есть логикой программ, но не программирования. Вопросы же корректных семантических спецификаций задач, семантико-синтаксического построения соответствующих программ, их верификации и т.д. в ней остаются открытыми. Интегрирующая роль ИПС заключается в том, что она ориентирована, в первую очередь, на изучение процессов построения программ через адекватное сути выделение и дальнейшее изучение общезначимых свойств программ. Одним из таких фундаментальных свойств семантических структур программ есть свойство редукционности. Оно адекватным образом вскрывает декомпозиционные структуры задач посредством, в частности, введения понятий регулярной редукции и $h$-редукции [5,6].

Функцию $g$ называют *регулярной редукцией* функции $f$ если справедливо следующее равенство $g \circ f = f$ [1], где $\circ$ - операция мультиплицирования, сопоставляющая упорядоченной паре функций $(g, f)$ новую функцию $g \circ f$, которая представляет собой последовательное выполнение исходных функций, обобщающее обычное произведение функций.

Парадигмная значимость данного понятия состоит в возможности непосредственного получения по редукционной спецификации задачи ее коннотативной спецификации. При этом различные реализации последней в той или иной форме[2] получаются уже автоматически с вытекающей из самого построения корректностью. Понятие регулярной редукции носит общезначимый характер. В этом смысле оно представляет собой концептуально-единое средство вскрытия логики самых разнообразных задач. Однако далеко не всегда его применение адекватно сути решаемой задачи. Примеры такой неадекватности проявляются уже на самых простейших уровнях. Решение данной проблемы сводится к обогащению потенциально открытой редукционной логики новыми видами редукций. Одним из таких обогащений является прямое обобщение понятия регулярной редукции – $h$-редукция.

Функцию $g$ называют *$h$-редукцией* функции $f$ если справедливо следующее равенство $g \circ f = f \circ h$, где $h$ – произвольная, но фиксированная функция.

Заметим, что непосредственно из определения следует, что если $h$ – тождественная функция, то $h$-редукция функции $f$ совпадает с регулярной редукцией этой функции.

Непосредственно из приведенных определений следует, что данные виды редукций являются адекватными денотативными уточнениями различных рекурсивных механизмов. Следовательно, денотативные спецификации задач, полученные с помошью этих редукций могут быть автоматически транслированы в ту или иную форму их представления. Таким образом, программирование решения той или иной задачи, с точки зрения логики этого процесса, может быть сведено к нахождению подходящей редукции, то есть к решению уравнения вида $x \circ f = f$ или $x \circ f = f \circ h$. При этом необходимо, конечно, помнить, что хотя уже аппарат регулярных редукций является экстенсионально полным в универсуме вычислимых функций, говорить об интенсиональной полноте его и даже общего редукционного аппарата в принципе невозможно. Это, в частности, означает, что, если для некоторой вычислимой функции теоретически и существует нетривиальная редукция одного из известных ее видов, то задача практического ее нахождения для этой функции может оказаться существенно сложнее исходной задачи. Поэтому наряду с развитием методов нахождения решений уравнений относительно известных видов редукций, таких, например, как приведенные выше, вскрытие новых видов редукций, в силу интенсиональной открытости общего редукционного аппарата, является актуальной задачей.

Проиллюстрируем использование метода редукций на конкретных примерах программирования решения задач численного анализа. Для начала рассмотрим задачу вычисления $\sqrt{x}$ с заданной точностью $\varepsilon$, где $x$ и $\varepsilon$ – положительные вещественные числа. Разрешающим свойством этой задачи является известный результат – последовательность $y_0, y_1, y_2, ..., y_n, ...$, где $y_0 = a, y_{i+1} = \frac{1}{2}(y_i + \frac{x}{y_i}), i = 0,1,2,...$, а $a$ – произвольное положительное вещественное число независимо от выбора $a$ сходится к $\sqrt{x}$. Заметим, что данное свойство

---

[1] Равенство понимается как равенство сущностей типа функций

[2] Например, в виде программы в некотором языке программирования.

является чисто семантическим и никак не связано с формой реализации решения данной задачи. Семантика программы решающей данную задачу в свою очередь может быть уточнена в виде именной функции $f$, сопоставляющей именному множеству вида $\{(u,x),(v,\varepsilon),(w,a)\}$ именное множество вида $\{(w,y_m)\}$, где $y_m$ – первый член упомянутой последовательности, для которого выполняется условие $\left|y_m^2 - y_{m-1}^2\right| < \varepsilon$. Таким образом, решение нашей основной задачи – спецификация логики программы вычисления $\sqrt{x}$, сводится к решению уравнения $x \circ f = f$. Очевидно, что решение его – есть редукция функции $f$. Отталкиваясь от разрешающего свойства задачи, рассмотрим именную функцию $g \equiv w_{pr} := w; w := \frac{1}{2}(w_{pr} + \frac{u}{w_{pr}})$, определенную на именных множествах вида $\{(w_{pr}, m),(w,s),(u,x)\}$. Легко убедиться, что именная функция $g$ является искомой редукцией. Действительно, она преобразует именное множество $\{(u,x),(v,\varepsilon),(w,a)\}$ в именное множество $\{(u,x),(v,\varepsilon),(w,\frac{1}{2}(a+\frac{x}{a}))\} = \{(u,x),(v,\varepsilon),(w,y_1))\}$. Но последнее именное множество под действием $f$ переходит, если конечно $\left|y_m^2 - y_{m-1}^2\right| \geq \varepsilon$, в то же самое именное множество $\{(w,y_m)\}$. Теперь коннотативная спецификация функции $f$ может быть получена автоматически:

$$f \equiv repeat\ \ g\ \ \ until\,|\,w^2 - w_{pr}^2\,| < v \equiv repeat\ \ w_{pr} := w; w := \frac{1}{2}(w_{pr} + \frac{u}{w_{pr}})\ \ until\ \ |\,w^2 - w_{pr}^2\,| < v.$$

Следующий пример иллюстрирует применение метода редукций для решения более общей, по сравнению с предыдущим, задачи. Рассмотрим класс специальных уравнений вида $x = \varphi(x)$, где функция $\varphi(x)$ удовлетворяет следующим двум условиям:

1. она определена и непрерывно дифференцируема на всей числовой прямой;
2. существует такое вещественное число $p < 1$, что для любого $x$ модуль производной $|\,\varphi'(x)| \leq p$.

Необходимо дать коннотативную спецификацию решения задачи поиска корней таких уравнений. Разрешающей особенностью задачи является то, что применительно к данному классу уравнений метод последовательных приближений сходится. Поэтому каждое уравнение из этого класса имеет единственное вещественное решение. Причём его можно найти начав с произвольного вещественного числа $x_0$ (начального приближения) и построив последовательность $x_0, x_1, x_2, \ldots$, где $x_i = \varphi(x_{i-1}), i = 1,2,3,\ldots$, сходящуюся к решению уравнения $x = \varphi(x)$. Таким образом, в основе решения лежит сведение поиска корня уравнения к вычислению его приближения – элемента последовательности приближений $x_n$, который удовлетворяет двум условиям:

1) для любого $i < n$ $\ |\,x_i - x_{i-1}| \geq \varepsilon$, где $\varepsilon$ – наперёд заданное положительное вещественное число, называемое точностью вычисления;

2) $|\,x_n - x_{n-1}| < \varepsilon$.

Исходная постановка задачи может быть уточнена, например, так: дать коннотативную спецификацию именной функции $f$, сопоставляющей именному множеству вида $\{(v,x_0),(u,\varepsilon)\}$ именное множество вида $\{(v,x_n)\}$, где $x_n$ – первый член последовательности приближений, удовлетворяющий условию 2. Очевидно, что в качестве редукции функции $f$ может быть выбрана именная функция $g \equiv v_{pr} := v;\ v := \varphi(v_{pr})$, определенная на именных множествах вида $\{(v_{pr}, m),(v,n)\}$, так как она адекватно отражает разрешительное свойство вычисления задачи. Таким образом, автоматически получаемая композиционная семантика функции $f$ задается следующим выражением:

$$f \equiv repeat\ \ g\ \ \ until\,|\,v - v_{pr}\,| < u \equiv repeat\ \ v_{pr} := v;\ v := \varphi(v_{pr})\ \ until\ \ |\,w^2 - w_{pr}^2\,| < v.$$

В отличие от предыдущего, здесь выражение задает коннотативную спецификацию бесконечного класса задач, что обусловлено наличием в выражении параметра (оракула) $\varphi$. В этом смысле, выражение задает уже не решение конкретной задачи, а его, хотя и простую (рудиментарную), но все же схему. Простота ее обусловлена тем, что использование оракула ограничивается уровнем тривиальной подстановки и при этом в выражении нет межоракульных зависимостей. Такого типа спецификации характерны для «условно-открытых» систем, в которых хотя и поддерживается возможность интеграции в них новых подсистем, однако только на уровне рудиментарного взаимодействия с другими подсистемами.

Наконец, рассмотрим пример построения процедурной спецификации схемы решения задач в их еще более общей постановке, характеризующейся не только наличием оракулов в такой спецификации, но и их нетривиальным взаимодействием. Решим вначале задачу вычисления функции, заданной операцией суммирования $f(x_1,...,x_{n-1},m)=\sum_{i=1}^{m} g(x_1,...,x_{n-1},i)$, где $g(x_1,...,x_{n-1},i)$ – произвольная, но фиксированная функция, зависящая от вещественных переменных $x_1,...,x_{n-1}$ и переменной $i$, принимающей натуральные значения. Таким образом, аналогично предыдущему необходимо дать процедурную спецификацмю именной функции $f$, преобразующей именные множества вида $\{(v_1,a_1),...,(v_{n-1},a_{n-1}),(v_n,m)\}$ в именные множества вида $\{(w, f(x_1,...,x_{n-1},m))\}$, где $a_1,...,a_{n-1}$ – вещественные числа, а $m$ – натуральное число. На этот раз решение исходной задачи – спецификация коннотативной логики именной функции $f$, сводится к решению уравнения вида $x \circ f = f \circ h$, то есть к нахождению ее $h$–редукции. Для этого обратимся к разрешающему свойству исходной функции $f(x_1,...,x_{n-1},m)$, которое выражается следующим рекурентным соотношением:

$$f(x_1,...,x_{n-1},1)=g(x_1,...,x_{n-1},1),$$
$$f(x_1,...,x_{n-1},m+1)= f(x_1,...,x_{n-1},m)+g(x_1,...,x_{n-1},m+1).$$

Легко удостовериться, что именная функция $p \equiv v_n := v_n + 1$ является $h$-редукцией функции *f*, где $h \equiv u := u+1; w := w+g$, а $g$ – именная функция, сопоставляющая каждому именному множеству $\{(v_1,a_1),...,(v_{n-1},a_{n-1}),(u,m)\}$ число $g(a_1,...,a_{n-1},m)$. Отсюда,

$$f \equiv u := 0;\ w := 0;\ repeat\ u := u+1;\ w := w+g;\ v_n := v_n \text{ -1 } until\, v_n \neq 0$$

или

$$f \equiv u := 0;\ w := 0;\ repeat\ \ u := u+1;\ w := w+g\ \ until\ \ v_n = u$$

Полученные схемы, как и предшествующая им, являются рудиментарными. Чтобы продемонстрировать возможности получения схем с нетривиальным взаимодействием оракулов несколько изменим исходную постановку задачи. Пусть теперь исходная функция задается следующим выражением: $f(x_1,...,x_n)=\sum_{i=1}^{m(x_1,...,x_n)} g(x_1,...,x_n,i)$, где $m(x_1,...,x_n)$ – функция вещественного аргумента и натурального значения, причем $m(x_1,...,x_n) \geq 1$ для любых значений $x_1,...,x_n$. В этом случае схема решения может быть, например, такой:

$f \equiv v_{n+1} := m;\ u := 0;\ w := 0;\ repeat\ \ u := u+1;\ w := w+g\ \ until\ \ v_{n+1} = u$, где $m$ – именная функция, преобразующая именные множества вида $\{(v_1,a_1),...,(v_n,a_n)\}$ в натуральные числа $m(a_1,...,a_n)$.

Для последней схемы характерна нетривиальная взаимосвязь, в данном случае, двух оракулов – именных функций $m$ и $g$. Очевидно, что уровень сложности таких взаимосвязей может быть сколь угодно высок. Это в общем случае означает необходимость вскрытия не только логики процессов получения коннотативных спецификаций на основе, например, метода редукций, но и логики получения и использования самих редукций. В этом смысле ИПС подразделяется на две взаимодополняюшие среды – среду микро- и макро-интеграции. Особую роль играет последняя, так как она позволяет непосредственно вовлечь в рассмотрения активную роль субъекта, значимость которой обстоятельно раскрыта в работах [12,13].

## Заключение

Системообразующей основой экспликации интеграционных средств, как и всей дескриптивной среды в целом является прагматико-обусловленный подход типизации универсума сущностей. Ядро такой системы составляет аппарат, адекватно поддерживающий интерфейс как между денотативными и коннотативными свойствами сущностей, так и между их общезначимыми и предметными (специальными) свойствами в целом. Рассмотренные нами интеграционная и предметно-интеграционная среды эксплицируают наиболее значимые свойства этого аппарата.

## Список литературы


1. Редько В. Н. Основания дескриптологии // Кибернетика и системный анализ. – 2003. – № 5. – С.16 – 36.
2. Редько В. Н. Основания программологии // Кибернетика и системный анализ. – 2000. - № 1. – С.35-57.
3. Редько В. Н. Дескриптологические основания программирования // Кибернетика и системный анализ. – 2002. – № 1. – С.3 – 19.
4. Редько И.В. Процесологічні середовища моделювання // Проблемы программирования.-2003.-№1.-С.37 – 48.
5. Редько В.Н., Гришко Н.В., Редько И.В. Эксплицативное программирование в среде логико-математических спецификаций// Труды Первой международной научно-практической конференции по программированию УкрПРОГ'98 (доклад). Киев. 1998. с.71-76.
6. Редько И.В., Гришко Н.В. Эксплицативное программирование в среде интеграции// Труды Первой международной научно-практической конференции по программированию УкрПРОГ'98 (доклад). Киев. 1998. с. 191-196.
7. Лейбниц Г. В. Сочинения в четырех томах.— М.: Изд-во «Мысль».—том 1.—1982.— 636 с.
8. Редько В. Н. Композиционная структура программологии // Кибернетика и системный анализ. – 1998. - № 4. - С.47-66.
9. Редько В. Н. Композиции программ и композиционное программирование // Программирование. – 1978. - № 5. – С.3-24.
10. Редько В. Н. Основания композиционного программирования // Программирование. – 1979. - № 3. С.3-13.
11. Редько В. Н. Семантические структуры программ. // Программирование.. – 1981. - № 1. С.3-19.
12. Вригт Г. Логико-философские исследования.–М: Изд-во «Прогресс».—1986.— 595 с.
13. Я. Хинтикка. Логико-эпистемологические исследования. — М.: «Прогресс».— 1980.— 447 с.